CptS 515 Advanced Algorithms HOMEWORK-1

Sharath Kumar Karnati

Question:4. Algorithms are to solve problems, or more precisely, to solve problems that have a precise mathematica definition. However, in practice, figuring out what is exactly the problem is not easy at all. (You may search internet) Suppose that you would like to write an algorithm to decide the similarity between two C programs. What would you do?

***Abstract*—In this paper, we are going to discuss different types of approaches to determine the similarity between two C programs. There are a finite number of ways that check the similarity between two programs. They can be categorized mainly into two types i.e. direct and indirect solutions, where indirect solutions the solution is more dependent on the source code of the codes, and in indirect solutions, the methodology is more dependent on core computer science topics like run time behaviour and by converting the sequence of events into graphs, chain, trees(data structures) and many more. The indirect solution method is preferred as the source code of the given codes will not be the same every time so it will be accurate to use indirect solution methods as it will convert the codes of both programs into identical data so we can easily verify their similarity. In this paper, we are going to discuss two indirect solution methodology where the similarity between two programs is guessed with the help of other methods that captures the structural information and syntactic features from the text data of codes. .**

I. Introduction

An algorithm is a process with a set of instructions to solve a specific problem. We can even create our own algorithm for a problem to achieve a solution in the way we want to be. If we want to solve a problem on our own, we must first understand the problem at hand by analyzing the deliverable as entirely or in parts. Every problem will definitely have a solution to it and we have to just develop a method and then try to resolve it. The aim is to build an algorithm that determines how similar two C programmes are to one another. When algorithms are created using techniques that are connected with one another, share the same inputs, and aim for the same outputs, they frequently resembles one another. Determining the scale of similarity will help us get closer to our goal in creation of the algorithm.

II. SOLUTIONS:

To find the similarity between two C programs there a many methods that are in exist which generally uses features like source code, complexity, output and inputs, etc from the given codes. But sometimes the given codes may be different in structural features mentioned above and can have a similarity, so it will be difficult task to check the similarity based on them.So, here we can use some methods in which the problem will be first assessed deeply and then provide a factual solution by having every provided variables involved and gives a perfect algorithm for the problem. In this paper, we are gonna mainly discuss about two methods called as:
1.White box comparison.
2.Black box comparison.

1.White box Comparison:

In this technique generally the source code of the two codes is available and will be categorized into trees , graphs ,sequences , tokens , data mining and machine learning.
Graphs: These are mathematical structures composed of nodes and edges that connect these nodes. Graphs can be directed or undirected. These are used to solve and model a wide range of problems, including social networks, transportation networks, recommendation systems etc.Graphs can have a complex structure with nodes connected in various ways so, they are not much preferred. Examples: Control Flow graphs, Scoring models etc.
Trees: These are hierarchical structures in data, such as directory structures, XML/HTML documents, and the abstract syntax tree(AST) in programming languages. Trees have a branching, hierarchical structure with a single root and multiple branches.They are easier to detect but have a high computational complexity. Examples: Sequence detection algorithm,AST etc

Tokens: These are the smallest units of a program or text.Tokens are used in lexical analysis,parsing and text processing tasks.They serve as the building blocks for complex data structures like trees and graphs.These have linear structure, as they appear in a sequential order within the text or code. Jacquard similarity, Liechtenstein distance based similarity, Suffix trees, XML similarity Detection Program Dependence Graph,etc are examples that are based on tokens , used to find the similarities.

Tokens, Trees and graphs serve different purposes in data representation and processing and their choice depends on the specific problem and the complexity of the data you need to handle.

2.Black Box Comparsion:

This technique is used for comparison when the source code of both the programs are unavailable. In this we generally turn the input or output into sequence of events and then see how similar they are to one and another. For example, if we compare a coffee maker and a washing machine there is no similarity in them such as the inputs ,outputs, functionality etc are all different then to compare these two machines first we have to convert them into a sequence as: Coffee maker : button press, sound , set temp button, high temp,..... Washing machine: button press, sound , select mode, select temp, set

temp,..... Then, we have to match the sequence and note them. Then we can use methods like Fourier transfer,Time series, Auto Correlation, or by using data structures like Marchov's Chain etc, to verify the similarity between the two sequences.
Time Series:Time series is nothing but data that refers to a sequence of data points that are collected or taken at specific intervals of time. Time series data is vastly used in various algorithms and applications of different domains as it has temporal nature and helps in making predictions , identifying different patterns and taking insights from different types of data.Here are some examples of time series in algorithms that we use in our daily life: Forecasting, Anomaly Detection, Pattern recognition, etc.

III. Preliminaries:

Now we are gonna see about different types of methods that can be used to find similarity between two programs using time series.To find similarity between two programs, particularly in a time series context, we have to compare their execution traces, code sequences or another program relevant characteristics. Time series techniques that can be used:

Dynamic Time warping(DTW):

DTW is a vastly used technique to calculate the similarity between two time series. We can represent program execution traces as time series and then apply DTW to compare them. This will be helpful to identify the similarities in order and timing of executed statements.

FCM clustering:

It is an unsupervised algorithm generally used in machine learning and data analysis. It is an extension of the traditional K -Means Clustering algorithm that generally allows data points to belong to multiple clusters with varying degrees of membership. FCM is specifically used to deal with data sets where data points may not belong exclusively to one cluster but also have some degree of association with multiple clusters.It will divide the data to clusters by determining the cluster centers and partition matrix. It comes with a concern of minimization problem but can overcome with Lagrange multiplier method.

Gravitational Search Algorithm :

The GSA is a optimization algorithm inspired by the law of gravity and the behaviour of celestial objects in the universe. The key concept behind GSA is that solutions with better fitness values act as attractive forces, pulling other solutions towards them. Over iterations the solutions converge towards optimal or near-optimal solutions.It is a population based algorithm so it can be applied for single -objective and multi-objective optimization

IV. Experiments and Results

Now we are gonna find similarity between two programs using time series.

Here the programs variables are converted into time series and produce to rectangular information granulation. The programs similarity is defined by the time series based on the information from the granulation .
The FCM clustering technique and the GSA are used to create the ideal rectangular information granules. The best rectangular information granules are used as input for the similarity measuring method between time series. The top and lower boundaries of the rectangular information granules are used to calculate the similarity between time series.
The geometry of spherical information granules, which is frequently employed, has been used in numerous fields. Compared to a rectangular information granule, the spherical one requires simply a core and a radius for description. The spherical information granule is more difficult to calculate than the rectangular one, especially when it comes to the cross area. The information granule's shape similarity should then be as straightforward as possible. Due to its straightforward structure, we generally opts for a rectangular information granule.
The geometry of spherical information granules, which is frequently employes, has been used in numerous fields. Compared to a rectangular information granule, The spherical ones requires simply a core and a radius for description. The spherical information granule is more difficult to calculate than the rectangular one, especially when it comes to the cross area.The information granule's shape similarity should then be as straightforward as possible. So, here we are gonna opt for rectangular information granule.
We are gonna use a 1- dimensional series here as to conveniently show the computation specifics because multidimensional time series are challenging specifics because multidimensional time series are challenging to understand and convey to others. Every time series dimension will be shown as an edge of the super rectangle.
The rectangular information granules of A and B are taken as inputs for a similarity calculation, which gives the similarity between the two programmes (A,B). The final time-series similarity between A and B is calculated by combining the obtained similarities for the two ideal numbers of information granules.
Rectangular information granule: The goal of rectangular information granulation of time series is to create rectangular information granules in the 2-D plane made up of time series data and its first order difference time series that are justifies and particular. It has two requirements to meet the criteria. Justifiability : As much original data as feasible must be accumulated within the boundaries of the produced information granule. It makes the information more valuable. Specificity: The smaller the information granule, the better it would as the created information granule should have a defined meaning.
Now we are gonna construct a 2-D plane by Y and Y, here time series is shown as set of points(yi, yi). The main aim is to construct a rectangular information granuale on a plane that

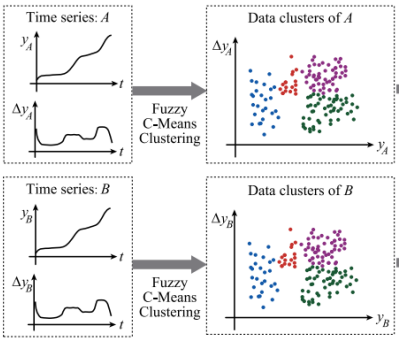
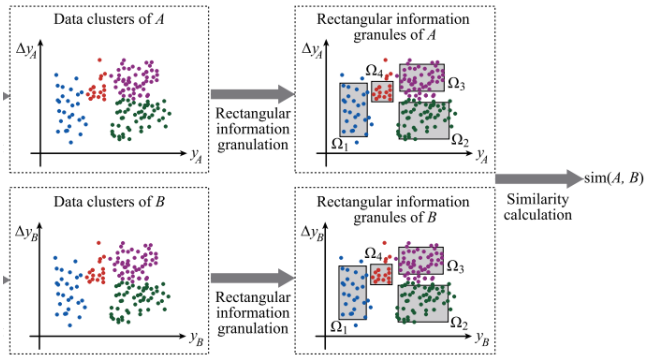


Fig. 1. Time-series similarity measurement scheme based on rectangular information granulation for two programs A and B



satisfies the justifiability as mentioned above. Here the median , min and max is given to constraints over Y and Y. The rectangular information granulation is an optimization problem. Now it has four decision variables that are used with the help of particle swarm optimization algorithm to develop information granules.But as GSA has a faster convergence speed and good global search capability than the swarm optimization algorithm which is used to produce more accurate information granules. The GSA is used for lower and upper bounds. Here multiple information granules are counted for change in time series,then FCM is used to provide the information on granule structure and the granules are developed into various C clusters, which then gives a partition matrix.

Measurement of Similarity: Here the similarity calculation is different from the traditional method, here it measures the similarity between time series from the context of rectangular information granulation where in traditonal way it is done as they find the distance between the pair of times series and calculates the similarity from that distances.

V. Scope of imnprovement:

We can use Fourier transform in place of time series as it will compare the similarities of the programs on their space and time complexity.

VI. Conclusion:

Here we have discussed about few methods to develop a algorithm to find the similarity between two programs without accounting on source code. The main thing in developing a algorithm is to first understand the problem and then converting it into a way which we have a strong knowledge on, so that we can easily find a solution. We can have a number of solutions once we get a clear understanding on the problem. Algorithms makes problem easier to understand but they are always trickier to develop. But once we figure out the algorithm the job becomes easier to find a trivial solution.

References

[1] https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=9664255
[2]https://ieeexplore.ieee.org/document/6915408
[3]https://ieeexplore.ieee.org/document/7726943
[4] https://www.sciencedirect.com/science/article/abs/pii/S0952197613001
[5]https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=9405994
[6]https://makeabilitylab.github.io/physcomp/signals/ComparingSignals/inc
[7] https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=9204963
[8]https://en.wikipedia.org/wiki/Cross-correlationCross-correlation
[9] https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=8488350